

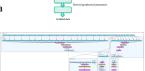
Scientific Workflow Engines for Earth Science Data Systems

9th ESDSWG: Technology Infusion Working Group Services, Interoperability, & Orchestration Subgroup

Hook Hua (JPL), Brian Wilson (JPL), Chris Lynnes (GSFC), MahaBaleshwa Hegde (GSFC)

What is a Workflow Engine?

- · Facilitates the flow of information, tasks,
- Provides method of orchestrating individual execution units
- · Management of control flow and data
- · Connects distributed models



· Codify production rules / policies

Science Data Processing

Workflow Patterns for Earth

Handling Large Data Transfers



- · Keep interface of workflow connections light Orchestration engine passes data location, and not the data itself
- Each service endpoint pulls in its own large input data

Configuration Not in Data Flow

- Configuration for each workflow component should not be in workflow pipes
- "lazy loading" of configuration Each workflow component reads configuration settings from file
- Enables modifications to configuration for long running workflow instances



Outdated Input Settings

- · Long runtimes of PGE
- · Need to check configuration inputs once PGE completed in case of change.
- · Rerun PGE workflow component if input configuration has changed

Vectorized Runs

Apply workflow on a sequence of data Example: Hyperspectral retrieval iterating through each pixel of image



Generic Software Architecture View



Workflow Language Expressivity

· Dataflow model / Entity-based

- The workflow is constructed from data processing and data transport (processors and data links).
- -Directed graphs
- Natural for scientific workflows
- -E.g. Simple Conceptual Unified Flow Language (Scufl)

· Process-centric model / Activity-based

Taverna SciFlo VisTrails Kepler Pegasus PHX ModelCenter Apache ODE

- -The nodes in the workflows are activities and the "data" passed between them form a control system rather than being a genuine flow of messages.
- "State transitions"
- Natural for business processes
- -E.g. Business Process Execution Language (BPEL)

Workflow Comparison

				1.0.1.2		WMS	ModelCenter		
About									
	Selling Points	mature, self-contained, gui editor	process migration, Python	visualization, provenance	extends Ptolemy II. workflows and custom-ized components can be saved, reused, and shared with colleagues using the Kepler archive format (KAR), integrated R statistical computing, direct access to EarthGrid.	can scale to a million tasks. Open Science Grid, TeraGrid, EC2 Cloud	trade space, design of experiments	support of WS-BPEL 2.0 CASES standard	workflow engine for Earth and space acience multi-missions
	References	http://www.taverna.org.uk/	http://scifio.jpl.nasa.gov/	http://www.vistralis.org/	https://kepler-project.org/	http://pegssus.isi.edu/	http://www.phoenix-int.com/ software/ phx_modelcenter.php	http://ode.apache.org/	http://ieeexplore.ieee.org/stamp stamp.jsp?smumber+04526650
General									
	Costs	open source	open source	open source	open source	open source	~\$30K	open source	open source
	License	LGPL		GPL	BSD	other	commercial	Apache Software License	based on JBOSS JBPM
	Maturity	production	production	in many projects	production	production	production		production
	User base	used in international mygrid collaborative portal	JPL.		multiple domains (ecology, biology, atrophysics)	multiple domains (astronomy, bioinformatics, earthquake acience, gravitational wave physics, ocean science, limnology, and others)	industry and academia, trade space design		JPL SDS production systems
	Project Owner	myGrid	JPL	University of Utah	UC Davis, UC Santa Barbara, and UC San Diego	USC Information Sciences Institute	Phoenix Integration	Apache	JPL
Development									
	Developer size	multi-institutional myGrid team and open source contributors	2-3 people	10+	multi-project contributors	7		10	٠
	Availability (years)	57	57	5				-3	5
	Footprint	small. Self-contained Java application	moderate dependencies	moderate	moderate	large. Grid dependencies	installable	small, relies on a Data Access Object implementation to provide persistent storage.	small
	Development language(s)	Java	Python	Python	Java	Java	binary install	Java JDKS	Java
Platform									
	Windows	yes	yes	yes	Yes	Yes	yes	Yes	yes
	MACOSX	yes	yes	yes	Yes	Yes	no no	Yes	yes
	Unus	yes	yes	yes	Yes	Yes	no no	Yes	yes
Usage									
	Integration effort Callable as API	moderate yes (Java)	moderate yes (Python)	moderate yex (Python)	moderate Yes (Java)	moderate Yes (Java)	moderate	moderate yes. Integration API.	moderate ves
						yes. Condor Schedd for task			
	Integrated Job Manager Browser Interface	no	no ves (AJAX)	no	no; a proposa exists for such a module yes (Hydrant)	management.	no	no	simple scheduler
Services	Browser Intertace		yes (AJAX)		yea (riyaran)				
	WSDL call augoort	VES	ves	ves	Yes			ves	no
	REST call support	ves	yes	yes	146			yes	no no
	LODI	,	,	,				,	no no
Workflow									
	Language	SCUFL	custom (XML)	custom (XML)	Custom (XML)		custom	DPDL	BPEL
	Visualization	GUI for visual job monitoring	yes (AJAX)	yes (VTK, pylab, and matplottib)	yes	Pegasus GUI for visual job monitoring	yes		simple logging
	Editor	GUI	yes (AJAX)	yes	yes	Pegasus GUI	yes		JIPM Editor
	Nesting Support	yes		yes	yes	Nested DAG			yes
Execution									
	Fault Tolerance	retries	retries		yes	error retries. "rescue workflows".		error recovery: uses transaction management.	flagging, restaging
		yes (xml)	yes	yes					
Extensibility	Caching	no no	yes		yes				
	Interface with Matlab	no	no	no	yes		yes		no
	Interface with IDL	100	100	100	no no		yes		no
	Interface with Python	no no	yes	yes	yes	yes	no no		no
	Interface with Eacel	ves (via Apache POI)	no	100	Yes (can be read by built-in		yes		no
Provenance		,			actor)				
	built-in support	under development		lineage for data products and processes	Yes	WingsPegasus Framework, semantic application-level proverance, execution provenance, tracks locations of data used and produced, and which software was used with which parameters.	no	process deployment versioning	no
	distributed external provenance								
	Open Provenance Model (OPM)	no	no	yes (translated from internal	Yes		no		no
	support Open Provenance Model Vocabulary (OPMV) support			provenance)					
	Vocabulary (OPWV) support ISO 19115 Lineage support								
-	naro narria cinnege Bupport								
Interoperablit									
,	Standards Supported	'horf'		OPM	Ptolemy II, OPM			WS-BPEL 2.0	BPEL.
Infusion									
	Projects Used in / Contact	myGrid eScience	Genesis / Brian Wilson «bdwlacn@pl.nasa.gov»		Giovanni / Christopher Lynnes «Chris Lynnes@nasa.gov»	Helio Seismology / http:// www.mps.mpg.de/projects/ seismo/GDC-SDO/		LIAN	JPL Missions: Phoenix, Diviner, MCS, MSL (soon) / Cecilis S Cheng
									rcecila a cheng@pl nasa gov>

Example Workflow Engines



- An open-source scientific workflow and provenance ms system developed at the University of Utah that provided data exploration and visualization. Emphasis on visualization and provenance





So Many Workflow Engines

 RUNA WFE Apache Agila
Apache ODE
Beexee JBoss jBPM SciFlo JFlower
JFolder • Bonita Syrup • Dalma Werkflow Modeling Workflow Engine (MWE)
Enhydra Shark
FlowMind Workpoin OpenWFE XFlow YAWL
Zebra

• Flux

Why So Many?

- · Domain-specific workflow features
 - -Data flow for Bioinformatics and Earth science
- -Activity flow for business process management
- · Fragmented "market"
- -Many derivatives of BPEL engines
- -Many custom adaptations
- · Popular workflow engines in each domainspecific field. Examples:
- -Kepler (ecology, Ptolemy II)
- -Taverna (biology)
- -VisTrails (visualization)
- -ModelCenter (DOE)

Where Are We At / Heading To?

- · Mixed results with workflow-based visual programming
- · Asynchronous services
- -WS-Eventing and WS-Messaging
- "Jobification" of SOAP/REST service interfaces
- · Integrating with other services - ServiceCasting, DataCasting, Federated OpenSearch, etc.
- · Collaborative workflows -myExperiment (Taverna)
- Semantic service and datatype ontology
- -ESIP and ESDSWG activity
- · Automated workflow discovery, execution, composition and interoperation
- -OWL-Services, WS-BPEL (legacy OASIS BPEL4WS)
- Open Provenance Model (OPM), semantic web services, and Proof Markup Language (PML)